```
RRRRRRRRRRRR      MMM         MMM      SSSSSSSSSSSS
RRRRRRRRRRRR      MMM         MMM      SSSSSSSSSSSS
RRRRRRRRRRRR      MMM         MMM      SSSSSSSSSSSS
RRR        RRR    MMMMMM    MMMMMM     SSS
RRR        RRR    MMMMMM    MMMMMM     SSS
RRR        RRR    MMMMMM    MMMMMM     SSS
RRR        RRR    MMM  MMM  MMM        SSS
RRR        RRR    MMM   MMM  MMM       SSS
RRRRRRRRRRRR      MMM         MMM        SSSSSSSSS
RRRRRRRRRRRR      MMM         MMM        SSSSSSSSS
RRRRRRRRRRRR      MMM         MMM        SSSSSSSSS
RRR    RRR        MMM         MMM              SSS
RRR    RRR        MMM         MMM              SSS
RRR    RRR        MMM         MMM              SSS
RRR      RRR      MMM         MMM              SSS
RRR      RRR      MMM         MMM              SSS
RRR      RRR      MMM         MMM              SSS
RRR        RRR    MMM         MMM      SSSSSSSSSSSS
RRR        RRR    MMM         MMM      SSSSSSSSSSSS
RRR        RRR    MMM         MMM      SSSSSSSSSSSS
```

```
RRRRRRRR   MM      MM   SSSSSSSS   IIIIII   DDDDDDD   XX      XX  LL          NN      NN  KK      KK
RRRRRRRR   MM      MM   SSSSSSSS   IIIIII   DDDDDDD   XX      XX  LL          NN      NN  KK      KK
RR    RR   MMMM  MMMM   SS           II     DD    DD   XX    XX   LL          NN      NN  KK    KK
RR    RR   MMMM  MMMM   SS           II     DD    DD   XX    XX   LL          NNNN    NN  KK   KK
RR    RR   MM MM MM MM  SS           II     DD    DD    XX  XX    LL          NNNN    NN  KK  KK
RR    RR   MM  MM  MM   SS           II     DD    DD     XXXX     LL          NN NN   NN  KKKKK
RRRRRRRR   MM      MM   SSSSSS       II     DD    DD      XX      LL          NN  NN  NN  KKKKK
RRRRRRRR   MM      MM   SSSSSS       II     DD    DD     XXXX     LL          NN   NNNN   KK  KK
RR  RR     MM      MM        SS      II     DD    DD    XX  XX    LL          NN    NNNN  KK   KK
RR   RR    MM      MM        SS      II     DD    DD   XX    XX   LL          NN      NN  KK    KK
RR    RR   MM      MM        SS      II     DD    DD   XX    XX   LL          NN      NN  KK    KK
RR    RR   MM      MM   SSSSSSSS   IIIIII   DDDDDDD   XX      XX  LLLLLLLLLL   NN      NN  KK    KK
RR    RR   MM      MM   SSSSSSSS   IIIIII   DDDDDDD   XX      XX  LLLLLLLLLL   NN      NN  KK    KK
```

```
RRRRRRRR      333333     222222
RRRRRRRR      333333     222222
RR    RR   33      33  22      22
RR    RR   33      33  22      22
RR    RR           33          22
RR    RR           33          22
RRRRRRRR           33          22
RRRRRRRR           33          22
RR  RR             33          22
RR  RR             33          22
RR    RR   33      33  22
RR    RR   33      33  22
RR    RR      333333   2222222222
RR    RR      333333   2222222222
```

        [ 2 0 1 , 1 0 ] R M S I D X L N K . R 3 2

        Define subroutine linkage

!++

! FACILITY:     RMS32 INDEX SEQUENTIAL FILE ORGANIZATION

! ABSTRACT:

!               This module defines all the routine linkage


! ENVIRONMENT:

!               VAX/VMS OPERATING SYSTEM

!--

! AUTHOR: D. H. Gillespie    CREATION DATE:  17-MAR-1978
!               and W. Koenig

! MODIFIED BY:

!       V03-024 RAS0154         Ron Schaefer            2-May-1983
!               Add NOPRESERVE (R2) to L_EXTEND0 linkage.

!       V03-023 MCN0020         Maria del C. Nasr        07-Apr-1983
!               Eliminage linkages of RMS$NULLKEY,and RMSCOMPRESS_KEY.
!               They will be using general linkages.  Modify L_ALLOC3,
!               and L_EXTEND0 to use parameters instead of global registers.

!       V03-022 MCN0019         Maria del C. Nasr        05-Apr-1983

```
         Preserve all registers except R0 and R1 in linkage
         FABREG.  RM$XSUMO requires a separate linkage because
         it cannot preserve R4.

   V03-021 TMK0001           Todd M. Katz            26-Mar-1983
         Add the linkage RABREG_4.

   V03-020 MCN0018           Maria del C. Nasr        24-Mar-1983
         Define new general linkages.  Also, since the linkages
         have changed so much, eliminate all history comments.

;**************************************************************
```

This module defines all the routine linkage for RMS-32 index file
organization.

KEEP THESE DEFINITIONS IN ALPHABETICAL ORDER PLEASE

The following conventions will be used for linkage macros:

```
        MACRO L_NAME =
                RL$NAME =
                JSB (REGISTERS) :
                GLOBAL (REGISTER DEFINITIONS) %;

        The register definitions are macros of the forms
                COMMON_FABREG, COMMON_RABREG, COMMON_IOREG, etc.
                or R_REGNAME as described in RMSIDXMAC.R32
```

MACRO

```
        L_ALDBUF =
                RL$ALDBUF =
                JSB (REGISTER = 5) :
                GLOBAL (R_IMPURE, R_IFAB)
                NOPRESERVE (2,3,4)
                NOTUSED (8,9) %,

        L_ALLOC3 =
                RL$ALLOC3 =
                JSB (REGISTER = 7; REGISTER = 1, REGISTER = 2) :
                GLOBAL (R_IFAB) %,

        L_BDBALLOC =
                RL$BDBALLOC =
                JSB (REGISTER = 4, REGISTER = 5) :
                GLOBAL (COMMON_RABREG)
                NOPRESERVE (2,3,4,5,6) %,

        L_CACHE =
                RL$CACHE =
                JSB (REGISTER = 1, REGISTER = 2, REGISTER = 3) :
                GLOBAL (COMMON_IOREG)
                NOPRESERVE (1,2,3)
                NOTUSED (8,9,10,11) %,

        L_CHECK_SEGMENT=
                RL$CHECK_SEGMENT =
                JSB (REGISTER = 0, REGISTER = 4, REGISTER = 2) :
                GLOBAL (R_IDX_DFN)
                NOPRESERVE (2,4,5)
                PRESERVE (1) %,

        L_CHKSUM =
                RL$CHKSUM =
                JSB (REGISTER = 5) :
                NOPRESERVE (0,1,2) %,
```

```
        L_COMPARE_KEY =
                RL$COMPARE_KEY =
                JSB (REGISTER = 1, REGISTER = 3, REGISTER = 0) :
                GLOBAL (R_IDX_DFN)
                NOPRESERVE (3) %,

        L_ERROR_LINK1 =
                RL$ERROR_LINK1 =
                JSB () :
                GLOBAL (COMMON_RABREG)
                PRESERVE (0) %,

        L_ERROR_LINK2 =
                RL$ERROR_LINK2 =
                JSB () :
                GLOBAL (COMMON_RABREG, R_IDX_DFN)
                PRESERVE (0) %,

        L_EXTEND0 =
                RL$EXTEND0 =
                JSB (REGISTER = 5, REGISTER = 6; REGISTER = 1, REGISTER = 6) :
                GLOBAL (COMMON_FABREG)
                NOPRESERVE (2,3,4,5) %,

        L_FABREG =
                RL$FABREG =
                JSB () :
                GLOBAL (COMMON_FABREG)
                NOPRESERVE (0,1) %,

        L_FABREG_7 =
                RL$FABREG_7 =
                JSB () :
                GLOBAL (COMMON_FABREG, R_IDX_DFN) %,

        L_GETSPC =
                RL$GETSPC =
                JSB (REGISTER = 1, REGISTER = 2; REGISTER = 1) :
                GLOBAL (R_IMPURE)
                NOPRESERVE (2,3,4)
                NOTUSED (8,9,10) %,

        L_JSB =
                RL$JSB =
                JSB () %,

        L_JSB01 =
                RL$JSB01 =
                JSB (REGISTER = 0, REGISTER = 1) :
                GLOBAL (R_BKT_ADDR, R_REC_ADDR, R_IDX_DFN, R_IRAB, R_IFAB)
                NOPRESERVE (0,1) %,

        L_LINK_7_10_11 =
                RL$LINK_7_10_11 =
                JSB () :
                GLOBAL (R_IDX_DFN, R_IFAB, R_IMPURE)
```

```
            NOPRESERVE (0,1) %,

    L_PRESERVE1 =
            RL$PRESERVE1 =
            JSB () :
            GLOBAL (COMMON_RABREG, R_BDB, R_REC_ADDR, R_IDX_DFN)
            PRESERVE (1) %,

    L_QUERY_AND_LOCK =
            RL$QUERY_AND_LOCK =
            JSB (REGISTER = 1, REGISTER = 2) :
            GLOBAL (COMMON_RABREG)
            NOPRESERVE (3) %,

    L_RABREG =
            RL$RABREG =
            JSB () :
            GLOBAL (COMMON_RABREG)
            NOPRESERVE (0,T) %,

    L_RABREG_4 =
            RL$RABREG_4 =
            JSB () :
            GLOBAL (COMMON_RABREG, R_BDB)
            NOPRESERVE (0,T) %,

    L_RABREG_4567 =
            RL$RABREG_4567 =
            JSB () :
            GLOBAL (COMMON_RABREG, COMMON_IOREG, R_REC_ADDR, R_IDX_DFN)
            NOPRESERVE (0,T) %,

    L_RABREG_457 =
            RL$RABREG_457 =
            JSB () :
            GLOBAL (COMMON_RABREG, COMMON_IOREG, R_IDX_DFN)
            NOPRESERVE (0,T) %,

    L_RABREG_467 =
            RL$RABREG_467 =
            JSB () :
            GLOBAL (COMMON_RABREG, R_BDB, R_REC_ADDR, R_IDX_DFN)
            NOPRESERVE (0,T) %,

    L_RABREG_567 =
            RL$RABREG_567 =
            JSB () :
            GLOBAL (COMMON_RABREG, R_BKT_ADDR, R_REC_ADDR, R_IDX_DFN)
            NOPRESERVE (0,T) %,

    L_RABREG_67 =
            RL$RABREG_67 =
            JSB () :
            GLOBAL (COMMON_RABREG, R_REC_ADDR, R_IDX_DFN)
            NOPRESERVE (0,T) %,
```

```
L_RABREG_7 =
        RL$RABREG_7 =
        JSB () :
        GLOBAL (COMMON_RABREG, R_IDX_DFN)
        NOPRESERVE (0,T) %,

L_REC_OVHD =
        RL$REC_OVHD =
        JSB (REGISTER = 1; REGISTER = 1) :
        GLOBAL (R_REC_ADDR, R_IDX_DFN, R_IFAB) %,

L_RELEASE =
        RL$RELEASE =
        JSB (REGISTER = 3) :
        GLOBAL (R_BDB, R_IRAB, R_IFAB, R_IMPURE)
        NOPRESERVE (1,2)
        NOTUSED (8) %,

L_RELEASE_FAB =
        RL$RELEASE_FAB =
        JSB (REGISTER = 3) :
        GLOBAL (R_BDB, R_IFAB, R_IFAB_FILE, R_IMPURE)
        NOPRESERVE (1,2)
        NOTUSED(8) %,

L_RETSPC =
        RL$RETSPC =
        JSB (REGISTER= 2, REGISTER = 3, REGISTER = 4) :
        GLOBAL (R_IMPURE)
        NOPRESERVE (2,3,5)
        NOTUSED (8,9,10) %,

L_SIDR_FIRST =
        RL$SIDR_FIRST =
        JSB (STANDARD; REGISTER = 1, REGISTER = 2) :
        GLOBAL (R_REC_ADDR, R_IDX_DFN, COMMON_RABREG) %,

L_XSUMO =
        RL$XSUMO =
        JSB () :
        GLOBAL (COMMON_FABREG)
        NOPRESERVE (0,T,4) %;
```

RMSIDXLNK
R32

RMS32MAC
MAR

UTLDEF
R32

RMSIDXMAC
R32

UTLDEFUND
R32

RMSIDXDEF
R32

NT0MACROS
MAR

RMSCALLS
MAR

RMS
LST

RMSMSCMAC
MAR

RMSINTDEF
LST